import requests

import datetime

from baidu\_id import province,city

def getIndex(word="我和我的祖国"):

"""

搜索指数

:param word:

:return:

"""

url = f"http://index.baidu.com/api/SearchApi/index?word={word}&area=0&days=30"

rep\_json = get\_rep\_json(url)

generalRatio = rep\_json['data']['generalRatio']

uniqid = rep\_json['data']['uniqid']

all\_index\_e = rep\_json['data']['userIndexes'][0]['all']['data']

pc\_index\_e = rep\_json['data']['userIndexes'][0]['pc']['data']

wise\_index\_e = rep\_json['data']['userIndexes'][0]['wise']['data']

t = getPtbk(uniqid)

startDate = rep\_json['data']['userIndexes'][0]['wise']['startDate']

all\_news = getTopNews(decrypt\_py(t, all\_index\_e), startDate, word)

pc\_news = getTopNews(decrypt\_py(t, pc\_index\_e), startDate, word)

wise\_news = getTopNews(decrypt\_py(t, wise\_index\_e), startDate, word)

for each in (all\_news, pc\_news, wise\_news):

print(each)

return None

def getFeedIndex(word="我和我的祖国"):

"""

:param word: 关键词

:return: 资讯指数

"""

url="http://index.baidu.com/api/FeedSearchApi/getFeedIndex?word=%s&area=0&days=30"%word

feed\_index\_data=get\_rep\_json(url)

uniqid=feed\_index\_data['data']['uniqid']

data=feed\_index\_data["data"]['index'][0]

generalRatio=data['generalRatio']#资讯指数概览

e=data['data']

t=getPtbk(uniqid)

return decrypt\_py(t,e)

def getNewsDate(word):

"""

:param word:

:return: 媒体指数的峰顶新闻

"""

url = f"http://index.baidu.com/api/NewsApi/getNewsIndex?area=0&word={word}&days=30"

res\_json = get\_rep\_json(url)['data']

generalRatio = res\_json["index"][0]['generalRatio']

e = res\_json['index'][0]['data']

start\_date = res\_json['index'][0]['startDate']

t = getPtbk(res\_json['uniqid'])

news=getTopNews(decrypt\_py(t, e),start\_date,word)

return news

def getTopNews(numList:list,start\_date,word):

"""

找到当前指数列表中的峰值

转换成日期字符串

将合成的日期字符串带入到请求数据接口中

返回新闻数据

:param numList: 指数列表

:param start\_date: 起始日期

:param word:

:return: 峰值新闻

"""

start\_date = string\_toDatetime(start\_date)

hill\_tops = getHilltop(numList)

hill\_tops\_date = [datetime\_toString(start\_date + datetime.timedelta(days=index)) for index in hill\_tops]

news = getNews(",".join(hill\_tops\_date), word)["data"][word]

return news

def getNews(dts,word):

"""

获取媒体指数接口数据

:param dts:用,连接的时间字符串，例：dts=2019-10-06,2019-10-10,2019-10-12,2019-10-16,2019-10-21,2019-10-24

:param word:

:return:接口传回的数据

"""

url=f"http://index.baidu.com/api/NewsApi/checkNewsIndex?dates[]={dts}&type=day&words={word}"

return get\_rep\_json(url)

def getHilltop(numList: list):

"""

:param numList:一组数值数组

:return:峰值的序号列表

"""

numList = list(map(lambda x: float(x) if x else 0, numList))

hillTops = [index for index, each in enumerate(numList) if

index and index < len(numList) - 1 and each > numList[index - 1] and each > numList[index + 1]]

return hillTops

def getMulti(word="我和我的祖国"):

"""需求图谱

pv搜索热度；ratio搜索变化率；sim相关性

"""

url=f"http://index.baidu.com/api/WordGraph/multi?wordlist%5B%5D={word}"

word\_data=get\_rep\_json(url)['data']['wordlist'][0]

if word\_data['keyword']:

print(word\_data['wordGraph'])

def getRegion(word="我和我的祖国",startDate='2019-09-17',endDate='2019-10-17'):

"""地域分布"""

url=f"http://index.baidu.com/api/SearchApi/region?region=0&word={word}&startDate={startDate}&endDate={endDate}"

region=get\_rep\_json(url)['data']['region'][0]

region\_city=[{'city':city[int(city\_n)],'number':region['city'][city\_n]}for city\_n in region['city']]

region\_prov=[{'prov':province[int(prov\_n)],'number':region['prov'][prov\_n]}for prov\_n in region['prov']]

print(region\_city,region\_prov)

def getBaseAttributes(word="我和我的祖国"):

"""人群属性"""

url=f"http://index.baidu.com/api/SocialApi/baseAttributes?wordlist[]={word}"

rep\_data=get\_rep\_json(url)['data']['result']

return rep\_data

def getInterest(word="我和我的祖国"):

"""兴趣分布"""

url=f"http://index.baidu.com/api/SocialApi/interest?wordlist[]={word}"

rep\_data = rep\_data=get\_rep\_json(url)['data']['result']

return rep\_data

def string\_toDatetime(string):

# 把字符串转成datetime

return datetime.datetime.strptime(string, "%Y-%m-%d")

def datetime\_toString(dt):

# 把datetime转成字符串

return dt.strftime("%Y-%m-%d")

def getPtbk(uniqid):

url=f"http://index.baidu.com/Interface/ptbk?uniqid={uniqid}"

return get\_rep\_json(url)['data']

def decrypt\_py(t,e):

"""

:param t:

:param e:

:return: 解析出来的数据

"""

a=dict()

length=int(len(t)/2)

for o in range(length):

a[t[o]] = t[length + o]

r="".join([a[each]for each in e ]).split(",")

return r

def get\_rep\_json(url):

"""

获取json

:param url: 请求接口

:return:

"""

hearder = {

"Cookie": 'BIDUPSID=84A5A425F78E7C3DE5BB7305B7B9EC39; PSTM=1542886758; BAIDUID=FB50141361C1DEF3ED1D3B52B9DAEDFD:FG=1; BDUSS=ZjfnVzdExheVBDZ1ozdm1VaG1yYnNXZHYwYmZIRjRvNXhZak9peHVpanVUWU5lRVFBQUFBJCQAAAAAAAAAAAEAAABY2FvRYWNnamtnAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAO7AW17uwFteUU; ZD\_ENTRY=baidu; delPer=0; PSINO=1; H\_PS\_PSSID=30962\_1445\_21086\_30905\_31085; BDORZ=FFFB88E999055A3F8A630C64834BD6D0; bdindexid=9pnhui45uvpc2o8p2c43bflmt7; Hm\_lvt\_d101ea4d2a5c67dab98251f0b5de24dc=1584947953,1585025541; Hm\_lpvt\_d101ea4d2a5c67dab98251f0b5de24dc=1585034857; RT="sl=0&ss=k85j4rb6&tt=0&bcn=https%3A%2F%2Ffclog.baidu.com%2Flog%2Fweirwood%3Ftype%3Dperf&z=1&dm=baidu.com&si=9kbgg6v3m99"'

#请填写游览器中的cookie

'User-Agent' 'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/537.36'

}

response = requests.get(url, headers=hearder)

response\_data = response.json()

print(response\_data)

return response\_data

def main():

getFeedIndex()

getNewsDate()

getIndex()

getRegion()

getBaseAttributes()

getInterest()

if \_\_name\_\_=="\_\_main\_\_":

main()

D:\python.exe: can't find '\_\_main\_\_' module in ''

[Finished in 0.4s]

我们还尝试使用GitHub上的一个爬虫代码，但没有成功，最后采用了另外的代码获取了百度搜索指数，不过还是很想利用上述代码获取媒体指数及咨讯指数

附GitHub代码：

from selenium import webdriver  
from selenium.webdriver.chrome.options import Options  
from config import COOKIES  
from urllib.parse import quote  
import datetime  
import time  
  
cookies = [{'name': cookie.split('=')[0],  
 'value': cookie.split('=')[1]}  
 for cookie in COOKIES.replace(' ', '').split(';')]  
  
chrome\_options = Options()  
chrome\_options.add\_argument('--headless')  
chrome\_options.add\_argument('--disable-gpu')  
browser = webdriver.Chrome(chrome\_options=chrome\_options)  
  
def init\_browser():  
 *"""  
 initialize browser  
 """* browser.get('https://index.baidu.com/#/')  
 browser.set\_window\_size(1500, 900)  
 browser.delete\_all\_cookies()  
 for cookie in cookies:  
 browser.add\_cookie(cookie)  
 browser.refresh()  
  
def get\_into\_page(keyword):  
 *"""  
 get baiduIndex page  
 """* url = 'https://index.baidu.com/v2/main/index.html#/trend?words=%s' % quote(keyword)  
 browser.get(url)  
  
def get\_time\_range\_list(startdate, enddate):  
 *"""  
 max 6 months  
 """* date\_range\_list = []  
 startdate = datetime.datetime.strptime(startdate, '%Y-%m-%d')  
 enddate = datetime.datetime.strptime(enddate, '%Y-%m-%d')  
 while 1:  
 tempdate = startdate + datetime.timedelta(days=300)  
 if tempdate >= enddate:  
 all\_days = (enddate-startdate).days  
 date\_range\_list.append((startdate, enddate, all\_days+1))  
 return date\_range\_list  
 date\_range\_list.append((startdate, tempdate, 301))  
 startdate = tempdate + datetime.timedelta(days=1)  
  
def ignore\_baidu\_index\_bug():  
 *"""  
 百度咨询指数，第一次点击时间会没有响应  
 """* time.sleep(1)  
 browser.find\_elements\_by\_xpath('//\*[@class="index-date-range-picker"]')[1].click()  
 base\_node = browser.find\_element\_by\_xpath('//\*[contains(@class, "index-date-range-picker-overlay-box") and \  
 contains(@class, "tether-enabled")]')  
 time.sleep(1.5)  
 base\_node.find\_element\_by\_xpath('.//\*[@class="primary"]').click()  
  
def adjust\_time\_range(startdate, enddate, kind):  
 *"""  
 ...  
 """* time.sleep(2)  
 browser.find\_elements\_by\_xpath('//\*[@class="index-date-range-picker"]')[kind].click()  
 base\_node = browser.find\_element\_by\_xpath('//\*[contains(@class, "index-date-range-picker-overlay-box") and \  
 contains(@class, "tether-enabled")]')  
 select\_date(base\_node, startdate)  
 end\_date\_button = base\_node.find\_elements\_by\_xpath('.//\*[@class="date-panel-wrapper"]')[1]  
 end\_date\_button.click()  
 select\_date(base\_node, enddate)  
  
 base\_node.find\_element\_by\_xpath('.//\*[@class="primary"]').click()  
 time.sleep(1)  
  
def select\_date(base\_node, date):  
 *"""  
 select date  
 """* time.sleep(2.5)  
 base\_node = base\_node.find\_element\_by\_xpath('.//\*[@class="right-wrapper" and not(contains(@style, "none"))]')  
 next\_year = base\_node.find\_element\_by\_xpath('.//\*[@aria-label="下一年"]')  
 pre\_year = base\_node.find\_element\_by\_xpath('.//\*[@aria-label="上一年"]')  
 next\_month = base\_node.find\_element\_by\_xpath('.//\*[@aria-label="下个月"]')  
 pre\_month = base\_node.find\_element\_by\_xpath('.//\*[@aria-label="上个月"]')  
 cur\_year = base\_node.find\_element\_by\_xpath('.//\*[@class="veui-calendar-left"]//b').text  
 cur\_month = base\_node.find\_element\_by\_xpath('.//\*[@class="veui-calendar-right"]//b').text  
 diff\_year = int(cur\_year) - date.year  
 diff\_month = int(cur\_month) - date.month  
 if diff\_year > 0:  
 for \_ in range(abs(diff\_year)):  
 pre\_year.click()  
 elif diff\_year < 0:  
 for \_ in range(abs(diff\_year)):  
 next\_year.click()  
  
 if diff\_month > 0:  
 for \_ in range(abs(diff\_month)):  
 pre\_month.click()  
 elif diff\_month <0:  
 for \_ in range(abs(diff\_month)):  
 next\_month.click()  
  
 time.sleep(1)  
 base\_node.find\_elements\_by\_xpath('.//table//\*[contains(@class, "veui-calendar-day")]')[date.day-1].click()  
  
def loop\_move(all\_days, keyword, kind):  
 *"""  
 to get the index by moving mouse  
 """* time.sleep(1)  
 chart = browser.find\_elements\_by\_xpath('//\*[@class="index-trend-chart"]')[kind]  
 chart\_size = chart.size  
 move\_step = all\_days - 1  
 step\_px = chart\_size['width'] / move\_step  
 cur\_offset = {  
 'x': step\_px,  
 'y': chart\_size['height'] - 50  
 }  
  
 webdriver.ActionChains(browser).move\_to\_element\_with\_offset(  
 chart, 1, cur\_offset['y']).perform()  
 yield get\_index(keyword, chart)  
  
 for \_ in range(all\_days-1):  
 time.sleep(0.05)  
 webdriver.ActionChains(browser).move\_to\_element\_with\_offset(  
 chart, int(cur\_offset['x']), cur\_offset['y']).perform()  
 cur\_offset['x'] += step\_px  
 yield get\_index(keyword, chart)  
  
def get\_index(keyword, base\_node):  
 *"""  
 get index datas by html  
 """* date = base\_node.find\_element\_by\_xpath('./div[2]/div[1]').text  
 date = date.split(' ')[0]  
 index = base\_node.find\_element\_by\_xpath('./div[2]/div[2]/div[2]').text  
 index = index.replace(',', '').strip(' ')  
 result = {  
 'keyword': keyword,  
 'date': date,  
 'index': index,  
 }  
 return result  
  
def main(keyword, startdate, enddate, kind=0):  
 *"""  
 :kind; int, 0:搜索指数, 1:咨询指数  
 搜索指数最早的数据日期为2011-01-01  
 咨询指数最早的数据日期为2017-07-03  
 """* init\_browser()  
 get\_into\_page(keyword)  
 if kind == 1:  
 ignore\_baidu\_index\_bug()  
  
 date\_range\_list = get\_time\_range\_list(startdate, enddate)  
 for startdate, enddate, all\_days in date\_range\_list:  
 adjust\_time\_range(startdate, enddate, kind)  
 for data in loop\_move(all\_days, keyword, kind):  
 yield data  
 browser.quit()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 for data in main('张艺兴', '2017-07-03', '2018-09-01', 0):  
 print(data)